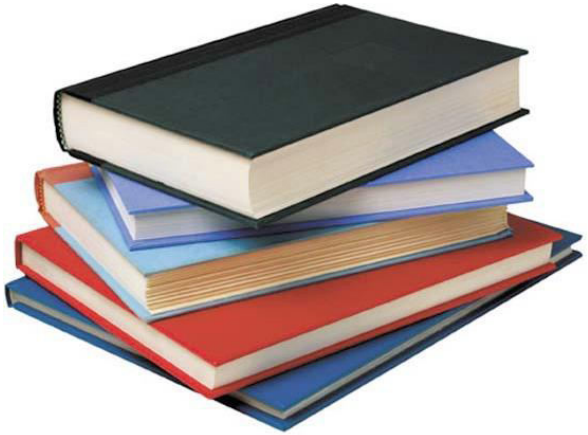
Stack

Farah Zakiyah Rahmanti

2014

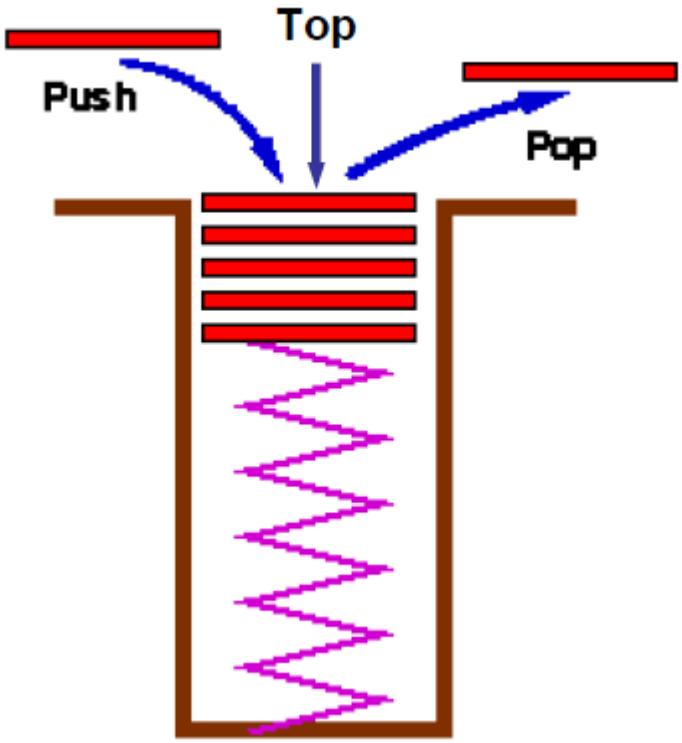


Description

- Effective concept to save and take data with rule “Last In First Out / LIFO”
- Data collection will be the opposite sequence with data storage

Push ()

- Push operation is an operation to add an element and placed on the top position of the stack



Example push ()

```
struct t_myStack {
    int data[100];
    int amount;
};

typedef struct t_myStack myStack;

myStack Stacksss;
```

```
void push(myStack* s, int myData)
{
    s->data[s->amount] = myData;
    s->amount++;
}
```

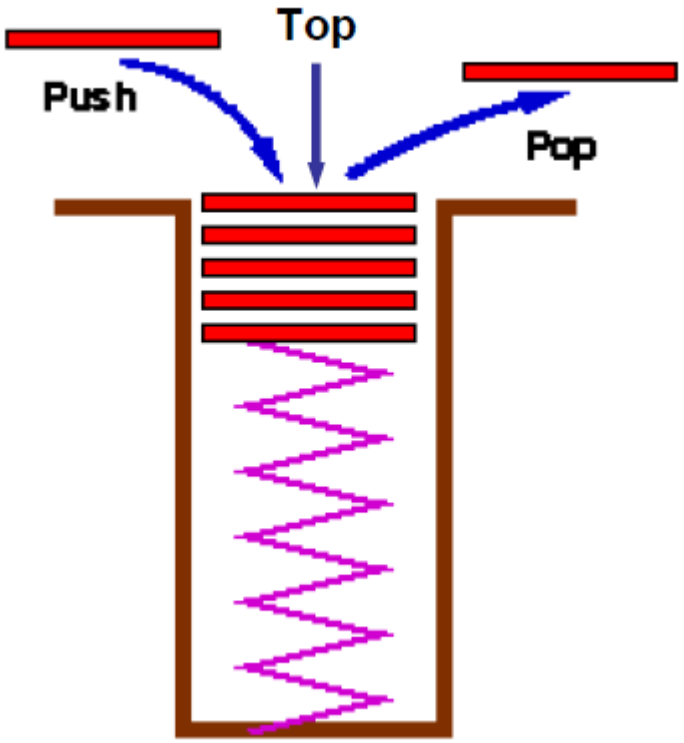
procedure **push()**

```
push(&Stacksss, 10);
push(&Stacksss, 20);
push(&Stacksss, 30);
```

Calling procedure **push()** of **main()**

Pop ()

- Pop operation is an operation to take an element which is placed on the top of stack



Example pop ()

```
struct t_myStack {
    int data[100];
    int amount;
};

typedef struct t_myStack myStack;

myStack Stacksss;
```

```
int pop(myStack* s)
{
    s->amount--;
    return s->data[s->amount];
}
```

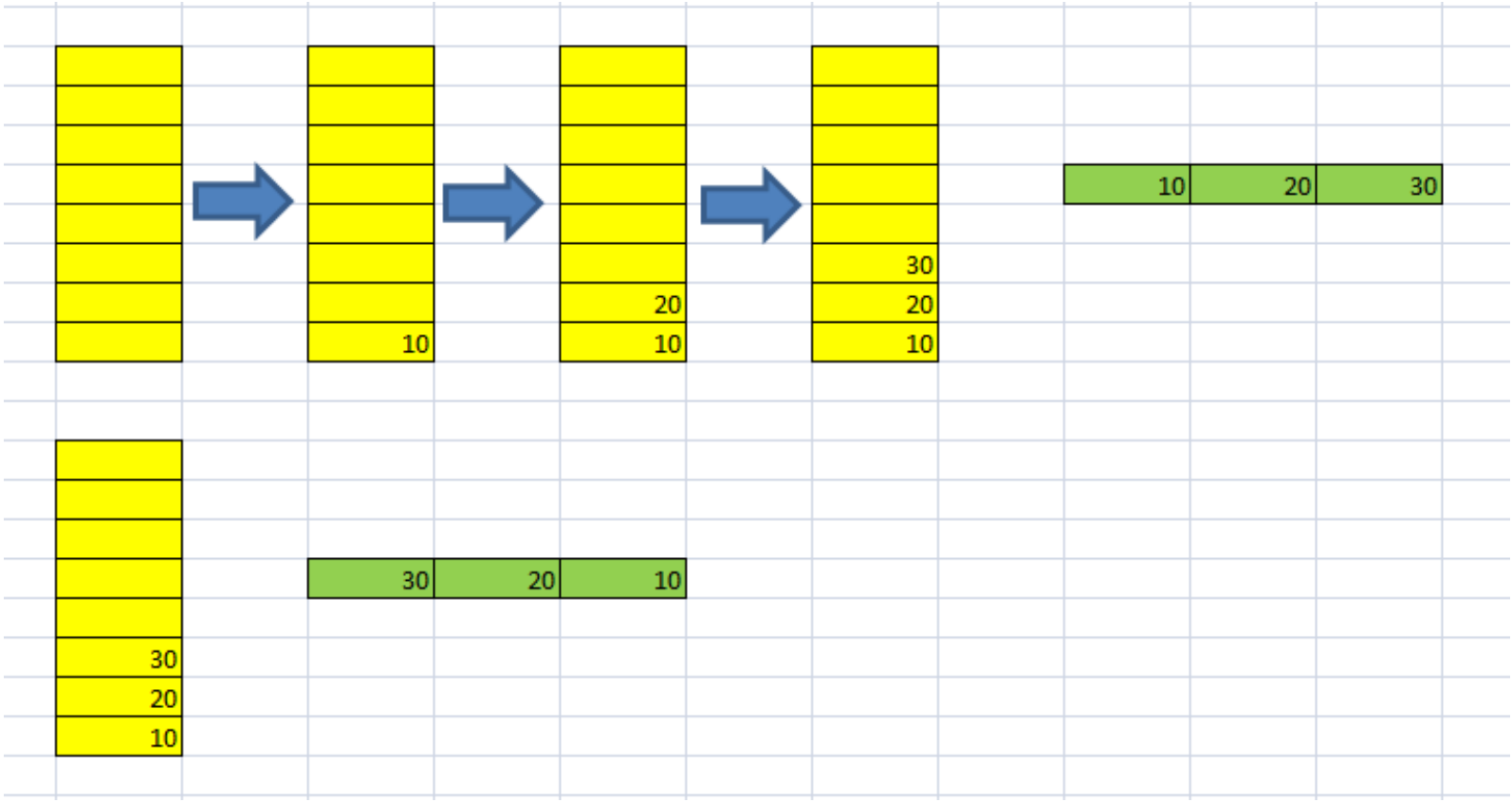
Function **pop()**

```
for(i=0; Stacksss.amount>0; i++)
    printf("%d ", pop(&Stacksss));
```

Calling function **pop()** of **main()**

Practicum

- Implement this stack !



Exercise

- Case study : mathematic expression evaluation
- Expression : in postfix notation
- Operator : *, /, +, -, ^

- Example :
- Input : AB*C/ || Output : (A*B)/C
- Input : ABC^/DE*+AC*- || Output : (A/(B^C))+(D*E)-(A*C)